目 录

第1章 EasyUDisk 使用手册 v1.0

1.1 概述

U 盘作为数据的载体目前已广泛应用于嵌入式行业，而某些电子产品在国家标准中已成了强制性的要求。在嵌入式的电子产品中实现对 U 盘文件的读写，成了许多厂家的技术攻关的难题。**优盘伴侣 EasyUDisk** 读写 U 盘模块就可以轻易解决以上问题，它是低端单片机操作 U 盘的最佳伴侣。

EasyUDisk 模块采用 PHILIP 公司工业级 USB 主控芯片 ISP1160 和 ARM7 微处理器 LPC2132，整个模块以工业级标准设计。模块提供了通用 RS232 串口与用户系统连接， 使普通的单片机也可以轻易实现对 U 盘的操作。

EasyUDisk 模块适用于仪器仪表、门禁系统、汽车黑匣子和税控机等对存贮速度要求不高的设备中使用。



模块特性：

- 全面支持大容量存储协议，真正支持所有非加密 U 盘。
- 支持 Bulk-Only 传输协议和 CIB-Only 传输协议。
- 支持 FAT12、FAT16 和 FAT32 文件管理系统。
- 提供标准 C 语言驱动源程序，适用所有单片机系统。
- 提供 RS232 电平接口，内部提供 TTL 电平接口。
- 支持 4800~115200 bps 标准波特率，上电后默认波特率为 9600 bps。

1.2 硬件连接

- 电源适配器

  输入： 9V 直流，400 mA

  极性： + ⊙ −

- 串口连接

DB-9 的 2 针为模块的 TXD；
3 针为模块的 RXD；
5 针为模块的 GND。

- **LED 状态指示**

绿色闪亮：待机状态，这时没有 U 盘插入；
绿色长亮：准备就绪，这时 U 盘已插入；
红色点亮：文件已打开，禁止拔出 U 盘；
红色熄灭：文件已关闭，可安全拔出 U 盘。

1.3 通讯协议

单片机系统通过串口与 EasyUDisk 模块交换数据，为了数据的可靠传输制定了以下简单的传输协议。为了更于描述，在下文中将单片机系统称为主机，EasyUDisk 模块称为从机。

主机与从机之间以帧方式传输数据，主机发出的为请求帧，而从机返回的为应答帧。请求帧和应答帧的数据结构相似，它们分别如表 1.1、表 1.2 所示。

- **串口参数**

4800，数据位 8，停止位 1，无奇偶校验，无流控制；
9600，数据位 8，停止位 1，无奇偶校验，无流控制；
14400，数据位 8，停止位 1，无奇偶校验，无流控制；
19200，数据位 8，停止位 1，无奇偶校验，无流控制；
38400，数据位 8，停止位 1，无奇偶校验，无流控制；
56000，数据位 8，停止位 1，无奇偶校验，无流控制；
57600，数据位 8，停止位 1，无奇偶校验，无流控制；
115200，数据位 8，停止位 1，无奇偶校验，无流控制；

- **请求帧**

请求帧为主机发出，帧头的两个字节固定为“U:”。从机接收到这两个字符开始接收帧的其它数据。

表 1.1 请求帧

字段	字节	说明
帧头	2 字节	字符“U:”。
功能代码	1 字节	请求执行操作代码。
参数	1 字节	指令参数，与具体指令相关。
数据长度	2 字节	数据区的长度，该域占 2 字节，低字节在前。
数据区	0~512	长度由数据域指出。

功能代码，是主机请求从机操作的指令。

参数，相关功能代码的基本参数，有些扩展参数通过数据区传输。

数据长度，指示后面数据区的长度，从机根据该域的数值，接收数据区的数据，接收到的数据字符个数与该域的值相等表示帧的结束。小端模式（低字节在前）。

数据区，其字符长度与数据长度域相关。当数据长度域值为 0，该帧则不包含数据区。

- 应答帧

应答帧，为从机响应主机的请求，并执行处理后返回的应答。

表 1.2 应答帧

字段	字节	说明
帧头	2 字节	字符“S:”。
功能代码	1 字节	与请求的功能代码一致。
状态	1 字节	功能代码执行后的状态，与具体指令相关。
数据长度	2 字节	数据区的长度，该域为 2 字节，低字节在前。
数据区	0~512	长度由数据域指出。

帧头的两个字节固定为“U:”。主机接收到这两个字符开始接收帧头后的数据。

功能代码，与请求帧的功能代码一致。

状态，从机处理功能代码的状态。正常完成返回 0，出错返回相关的功能代码。

数据长度，指示后面数据区的长度，主机根据该域的数值，接收数据区的数据，接收到的数据字符个数与该域的值相等表示帧的结束。小端模式（低字节在前）。

数据区，其字符长度与数据长度域相关。当数据长度域值为 0，该帧则不包含数据区。

- 无效帧

如果请求帧的头两个字符不为“U:”和应答帧的头两个字符不为“S:”为无效帧。

在一帧中字符传输间隔大于 1000 毫秒，为无效帧，并丢掉已接收的数据。

1.4 功能代码

EasyUDisk 模块支持的功能代码如表 1.3 所示。

表 1.3 功能代码表

名称	代码	说明
查询系统状态	0x00	获取系统相关信息，如系统时间、U 盘状态、已打开文件信息和软件版本号等。
系统设置	0x01	设置从机波特率和系统实时时钟。
打开文件	0x10	以读写方式打找，如果文件不存在将创建该文件。
关闭文件	0x12	文件打开后必需使用该请求关闭文件，否则数据可能丢失。
写文件	0x20	向打开的文件写入数据。
读文件	0x30	在打开的文件读出数据。
移到指针	0x40	移动读写文件中的指针。
创建目录	0x50	在指定的目录下创建目录。

1.4.1 查询系统状态

查询系统状态请求帧的数据结构，如表 1.4 所示。

表 1.4 查询系统状态请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x00
参数	1 字节	保留，写入 0x00
数据长度	2 字节	0x0000 (小端)
数据区	0	无

从机执行查询系统状态指令后返回应答帧的数据结构，如表 1.5 所示。

表 1.5 查询系统状态应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x00
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0014（小端）
波特率编码	1	值为 1~7。0：4800，1：9600，2：14400，3：19200，4：38400，5：56000，6：57600，7：115200
系统时钟	1	年
	1	月
	1	日
	1	时
	1	分
	1	秒
下行口状态	1	0，U 盘不存在；1，有一个 U 盘
文件 ID	1	不为 0xff 即为有文件打开
文件大小	4	小端，没有打开的文件或文件大小为 0，该域值为 0x0000。
固件版本号	1	01

续表 1.5

1.4.2 系统设置

系统设置请求帧的数据结构，如表 1.6 所示。上电后 EasyUDisk 默认使用的波特率为 9600 bps，系统实时钟为 05 年 09 月 11 日，12 点 00 分 00 秒。

由于对文件操作通常需要记录当前文件操作的时间，所以在对文件操作之前应先设置 EasyUDisk 的系统时间与当前时间一致。

表 1.6 系统设置请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x01

参数	1 字节	保留，写入 0x00
数据长度	2 字节	0x0007（小端）
波特率编码	1	值为 0~7。0：4800，0：4800，1：9600（上电默认），2：14400，3：19200，4：38400，5：56000，6：57600，7：115200
系统时钟	1	年（上电默认值为：05）
	1	月（上电默认值为：09）
	1	日（上电默认值为：11）
	1	时（上电默认值为：12）
	1	分（上电默认值为：00）
	1	秒（上电默认值为：00）

从机执行系统设置指令后返回应答帧的数据结构，如表 1.7 所示。

表 1.7 系统设置应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x01
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0000（小端）
数据区	0	无

1.4.3 打开文件

打开文件请求帧的数据结构，如表 1.8 所示。

表 1.8 打开文件请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x10
参数	1 字节	保留，写入 0x00
数据长度	2 字节	文件名字符长度
数据区	0~128 字节	文件夹\文件名

数据区为文件名字符，可包含文件夹的绝对路径。例如打开根目录的文件 EasyDisk.TXT 数据区如：“a:\ EasyDisk.TXT”，数据长度值为 15；打开“host”文件夹中的 usb.txt 文件数据区如：“a:\host\usb.txt”，数据长度为 15。

注意：当前版本不支持长文件名，文件名长度为 8 个字符，3 个字符扩展名。

从机打开文件请求完成后返回应答帧数据结构，如表 1.9 所示。

表 1.9 打开文件应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x10
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0005
数据区	4 字节	文件长度，小端
	1 字节	打开文件的 ID

文件成功打开状态代码为 0，数据长度为 5，数据区的前 4 个字节为打开文件的大小（为小端模式表示，第 1 个字节为低 8 位，依次类推），第 5 个字节为打开文件的编号，以后对该文件的操作都是需要使用该编号。

当前版本只支持同时打开一个文件。

1.4.4 关闭文件

关闭文件请求帧的数据结构，如表 1.10 所示。执行该请求可以关闭所有已打开的文件。

表 1.10 关闭文件请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x12
参数	1 字节	保留，写入 0x00
数据长度	2 字节	0x0000（小端）
数据区	0 字节	无

关闭文件应答帧的数据结构，如表 1.11 所示。

表 1.11 关闭文件应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x12
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0000（小端）
数据区	0 字节	无

1.4.5 写文件

写文件请求帧的数据结构，如表 1.12 所示。

表 1.12 写文件请求帧

字段	字节	说明

帧头	2 字节	字符“U:”
功能代码	1 字节	0x20
参数	1 字节	已打开文件的 ID
数据长度	2 字节	1~512
数据区	1~512 字节	写入数据长度

需输入的参数为已打开文件的 ID，通过打开文件请求可获得该 ID。数据长度为需要写入文件的字节数，数据区为需要写入的数据。数据的长度最大值为 512 字节。

写文件请求应答帧的数据结构，如表 1.13 所示。

表 1.13 写文件应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x20
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x00004（小端）
数据区	4 字节	实际写入的数据长度（小端）

数据长度固定为 4 字节，数据区用 4 个字节返回实际写入数据的字节数。并以小端模式保存返回实际写入文件的字节数，即数据区的第 1 个字节为该参数的低 8 位，依次类推。

当状态代码不为 0x00 时，数据区的参数无效。

1.4.6 读文件

读文件请求帧的数据结构，如表 1.14 所示。

表 1.14 读文件请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x30
文件 ID	1 字节	已打开文件的 ID
数据长度	2 字节	0x0004（小端）
数据区	4 字节	1~512 读出的数据长度（小端）

需输入的参数为已打开文件的 ID，通过打开文件请求可获得该 ID。数据长度固定为 4 并以小端模式保存。数据区为需要读出文件的字节数最大值为 512 字节，并以小端模式保存该参数。

读文件请求应答帧数据结构如表 1.15 所示。

表 1.15 读文件应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x30
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0000~0x0200（小端）
数据区	0~512	读出的数据

数据长度为读文件请求实际读出的数据长度，数据区为读出的数据。数据长度的值为 2 字节，该值的最大有效值为 0x200，当读请求出错时，该值为 0；数据区字符数与数据长度域的值一致。

1.4.7 移动指针

移动指针请求帧的数据结构，如表 1.16 所示。

表 1.16 移动指针请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x40
参数	1 字节	文件 ID
数据长度	2 字节	0x0005（小端）
数据区	4 字节	偏移量（小端）
	1 字节	转移类型（0：从文件头计算，1：从当前位置计算，2：从文件尾计算）

需要操作文件的 ID，通过打开文件请求可获得该 ID。数据长度为 5，数据区中包含两个参数：前 4 个字节为偏移量，第 5 个字节为转移类型。偏移量是 1 个 32 位的有符号整数；转移类型有三种方式，从文件的第 1 个字节开始，从文件的最后 1 个字节开始和从当前指针开始。

从文件的第 1 个字节开始转移，偏移量的取值范围为 0 到文件的结束；

从文件的最后 1 个字节开始转移，偏移量的取值范围为 0 到文件的开始，因为是从反方向转移所以该值为负数；

从当前指针开始转移，偏移量的取值范围分为两段：0 到文件的结束和 0 到文件的开始。0 到文件的开始为反向转移，该值使用负数表示。

移动指针请求应答帧数据结构如表 1.17 所示。

表 1.17 移到指针应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x40
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0004（小端）
数据区	4 字节	当前指针在文件的偏移

数据长度为 4，数据区返回的 4 个字节数据为当前指针在其文件的偏移值。数据类型为无符号整数，以小端模式保存。

1.4.8 创建目录

创建目录请求帧的数据结构，如表 1.18 所示。

表 1.18 创建目请求帧

字段	字节	说明
帧头	2 字节	字符“U:”
功能代码	1 字节	0x50
参数	1 字节	保留，写入 0x00
数据长度	2 字节	目录名字符长度（小端）
数据区	1~120	目录名字符数据

数据区为文件夹名字符，可包含文件夹的绝对路径。例如在根目录下创建文件夹（目录）名称为“USB”，数据区如：“a:\USB”，数据长度值为 6；在 USB 文件夹下创建文件夹“UDISK”，数据区如：“a:\USB\UDISK”，数据长度为 12。

注意：当前版本不支持长文件夹名，最长为文件夹名为 12 个字符，命名规则与 DOS 相同。

创建目录（文件夹）请求应答帧的数据结构，如表 1.19 所示。

表 1.19 创建目录应答帧

字段	字节	说明
帧头	2 字节	字符“S:”
功能代码	1 字节	0x50
状态代码	1 字节	正常为 0，出错返回非 0 值。
数据长度	2 字节	0x0000（小端）
数据区	0	无数据返回

目录创建成功则返回 0，出错返回非 0 值。如果创建目录已存在也返回非 0 值。